

Transmeta's Crusoe: Cool Chips for Mobile Computing

**David R. Ditzel
Chief Executive Officer
Transmeta Corporation**

Transmeta, Crusoe, Code Morphing and LongRun are trademarks of Transmeta Corporation

Transmeta Hot Chips Presentation - August 2000

Agenda

Transmeta's Crusoe Technology

Crusoe Microprocessors

Introduction of a new Crusoe processor

Hardware support features for Dynamic Translation

A few Crusoe Systems

What are the big problems for designers of Mobile Computers that need to talk to the Internet?

Heat

**High performance processors take 10 to 20 watts
Makes battery life unacceptably short.**

Compatibility with the Internet

**Need x86 compatibility to run
PC Software**

Web based software - such as macromedia flash

Browser plug-ins (plug-ins are tiny x86 programs)

Performance

Need PC desktop levels of performance for good experience

Transmeta's Crusoe Chip can solve these problems

Crusoe is a Family of Mobile Internet Processors

Low Power - for long battery life -- no fans

Compatible - with all x86 PC software

High Performance - for Internet applications

- **Streaming video (eg MPEG-4)**
- **Macromedia Flash**

How is Crusoe able to achieve these goals?

Crusoe
is the first
microprocessor
whose instruction set
is implemented entirely with
Software

Transmeta's Vision: A Software Based Microprocessor

- ◆ **New Idea: software could be an integral part of a microprocessor**
- ◆ **A combined hardware/software solution could have many benefits**
 - ◆ Simpler hardware chips
 - ◆ Easier to design and debug chips
 - ◆ Smaller design teams with shorter design times
 - ◆ No worry about backward compatibility in hardware
 - ◆ Less costly to manufacture smaller chips
 - ◆ Simpler chips would run cooler

**Plus software could LEARN as it ran –
the first SMART processor**

Crusoe Technology and Benefits

Crusoe is the sum of

Code Morphing Software

+



=

Code Morphing Software

- ◆ Dynamic x86 to VLIW translator
- ◆ Software optimized execution
- ◆ Learns and improves with time

VLIW Processor

- ◆ 128-bit Very Long Instruction Word
- ◆ Simple and fast Engine
- ◆ Significant reduction in transistors

Low
Power

x86
Compatibility

PC
Performance

$\frac{3}{4}$

+

$\frac{1}{4}$

Conventional HW vs Crusoe HW+SW

Conventional x86 Silicon Hardware	Crusoe	
	VLIW Silicon Hardware	Code Morphing Software
variable length instruction decode superscalar grouping logic superscalar issue logic bypass logic register renaming complex addressing modes out-of-order execution speculative execution arithmetic functions register files microcode ROM caches fp stack logic	simple decode in-order execution arithmetic functions register files caches	x86 decoding instruction grouping instruction scheduling bypass scheduling register renaming address mode synthesis speculative execution software libraries fp stack code optimization

The First Two Crusoe Processors



TM5400

Lightweight Notebook Computers
Microsoft Windows

- ◆ 700 MHz
- ◆ 400 KBytes of cache
- ◆ 1 watt
- ◆ x86 compatible



TM3200

Mobile Internet Appliances
Mobile Linux

- ◆ 400 MHz
- ◆ 108 Kbytes of cache
- ◆ 1 watt
- ◆ x86 compatible

TM5600: A New Crusoe Processor

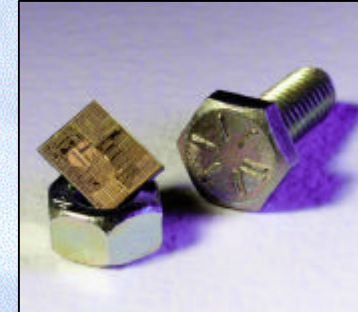
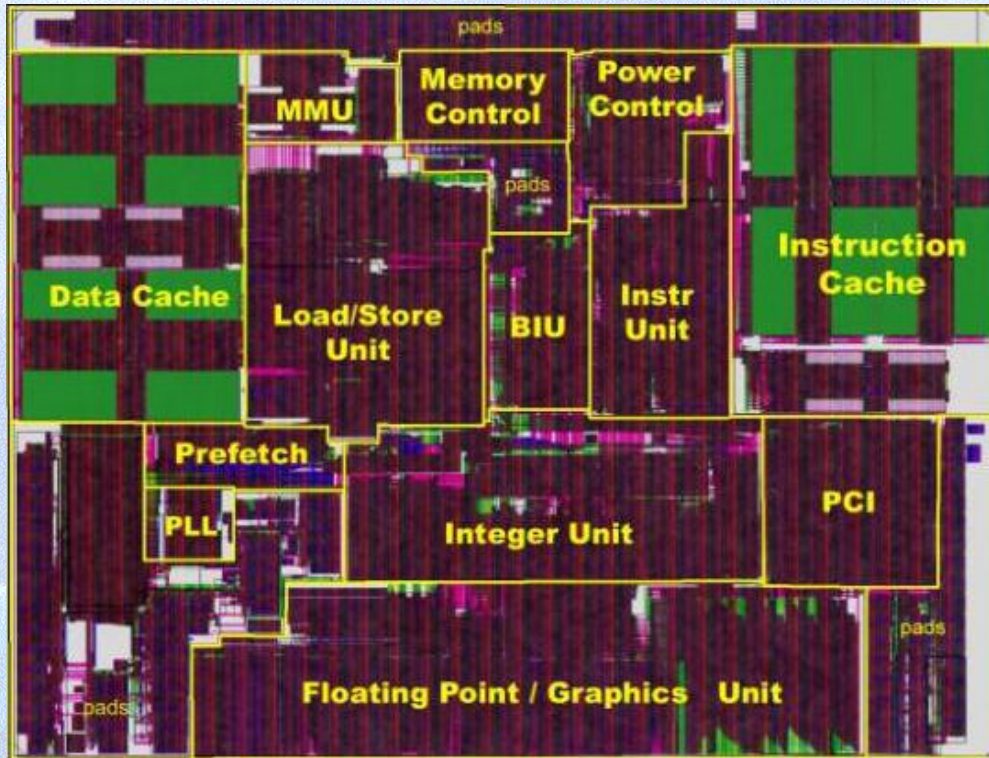


TM5600

**Lightweight Notebook Computers
Microsoft Windows**

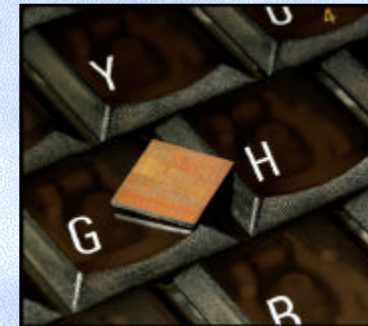
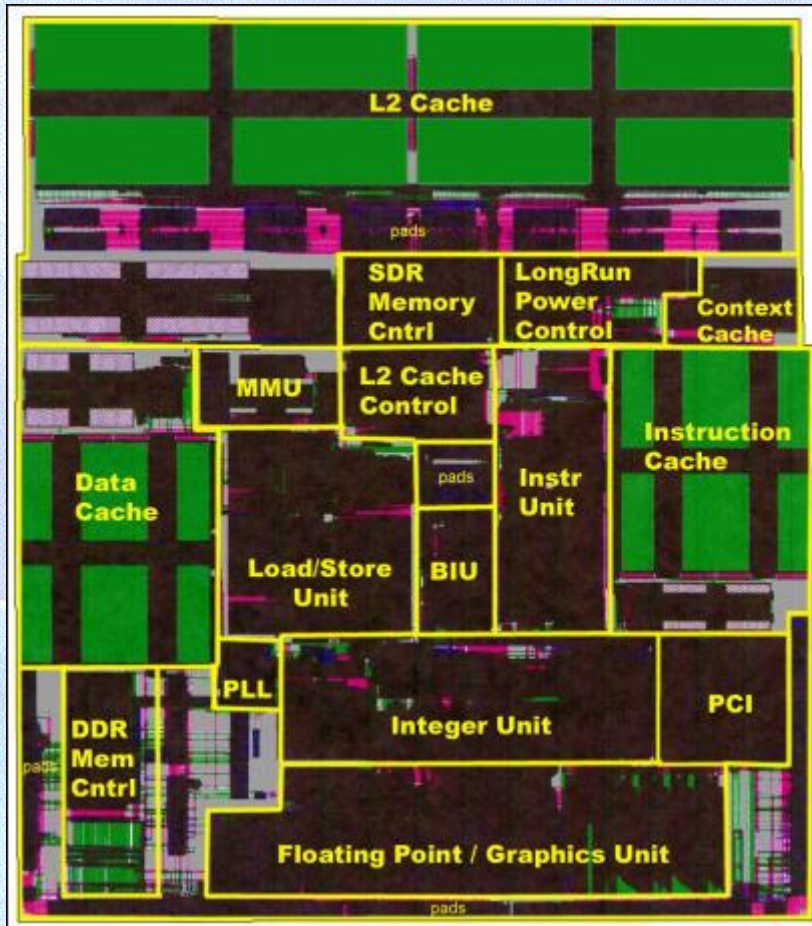
- ◆ **New TM5600 Crusoe Processor**
- ◆ **Same package pinout as TM5400**
- ◆ **700 MHz operation with LongRun**
- ◆ **656 KBytes of on-chip cache**
- ◆ **L2 increased to 512K bytes**
- ◆ **Performance increases by ~20% at same MHz**
- ◆ **Power reduced by ~10%**

TM3200 for Mobile Internet Devices



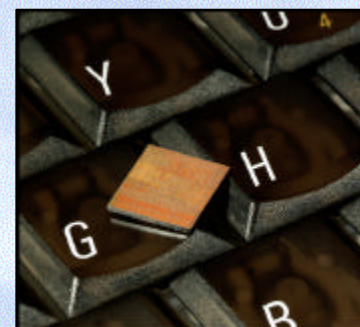
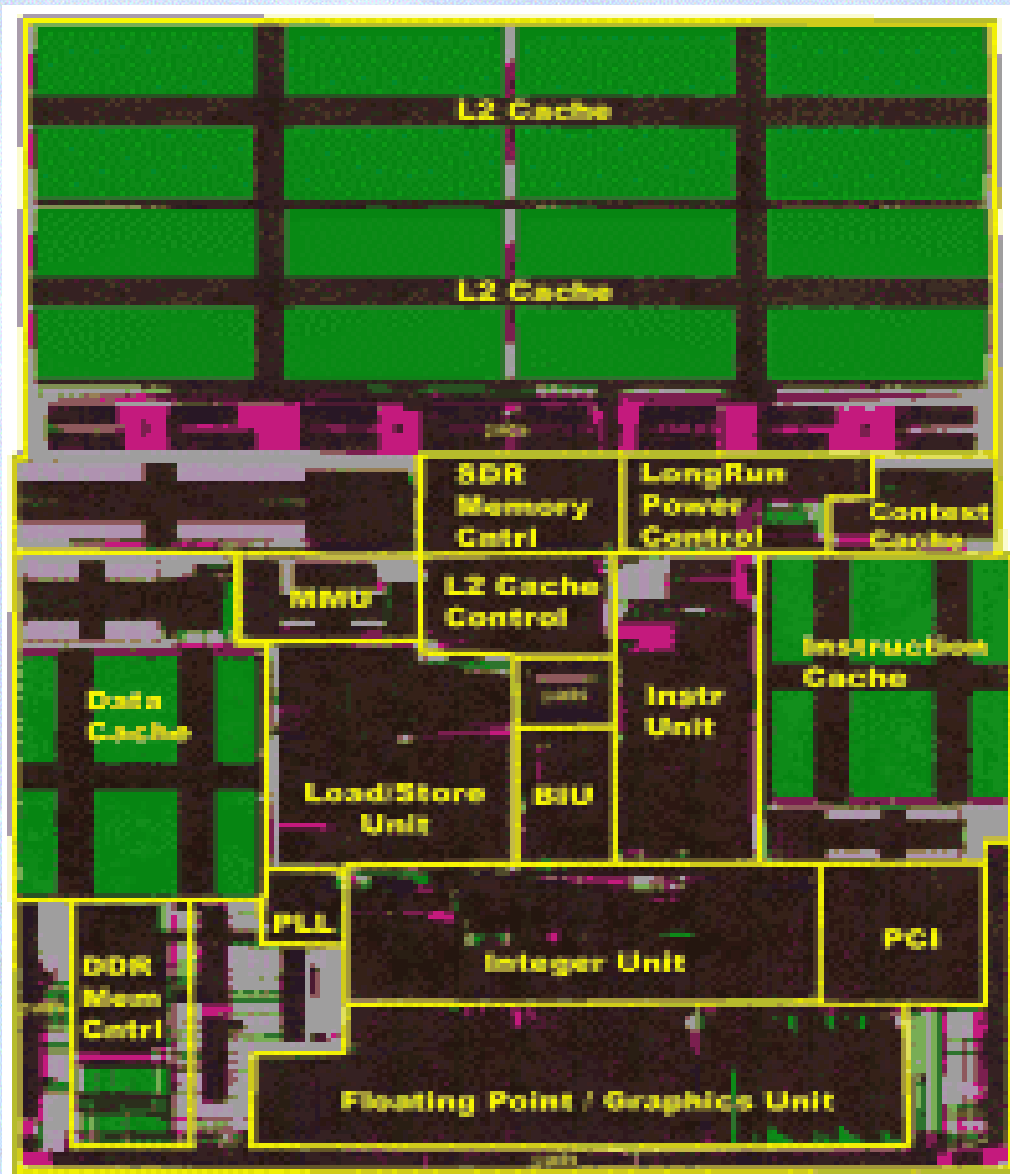
<u>TM3200</u>	
Frequency Range	333-400 MHz
L1 Cache	96KB
L2 Cache	
Main Memory	PC133 SDRAM
Upgrade Memory	
North Bridge	Integrated
Package	474 BGA
Fab Partner	IBM
Process Technology	.22u
Die Size	77 mm ²
Sample	Now
Production	Now

TM5400 for 2-4 Lb. Ultra-Light PCs



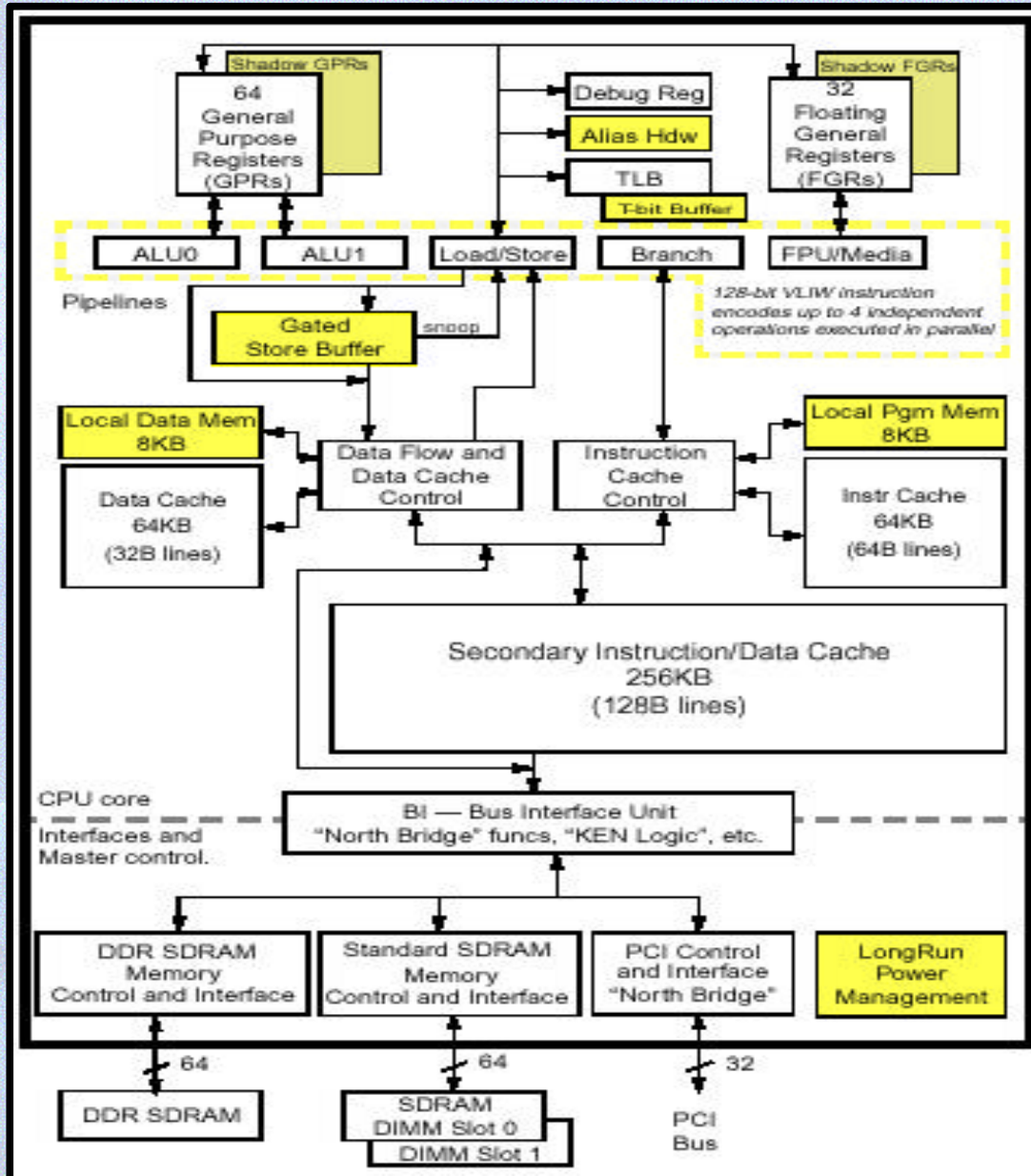
<u>TM5400</u>	
Frequency Range	500 – 700 MHz
L1 Cache	128K
L2 Cache	256K
Main Memory	DDR or SDRAM
Upgrade Memory	SDRAM
North Bridge	Integrated
Package	474 BGA
Fab Partner	IBM
Process Technology	.18u
Die Size	73mm ²
Sample	Now
Production	Now

TM5600 for 2-4 Lb. Full Featured Notebooks



TM5600	
Frequency Range	500 – 700 MHz
L1 Cache	128K
L2 Cache	512K
Main Memory	DDR or SDRAM
Upgrade Memory	SDRAM
North Bridge	Integrated
Package	474 BGA
Fab Partner	IBM
Process Technology	.18u
Die Size	88 mm ²
Sample	Now
Production	Now

Crusoe VLIW Processor Architecture



Free from x86 Legacy

128 bit VLIW processor

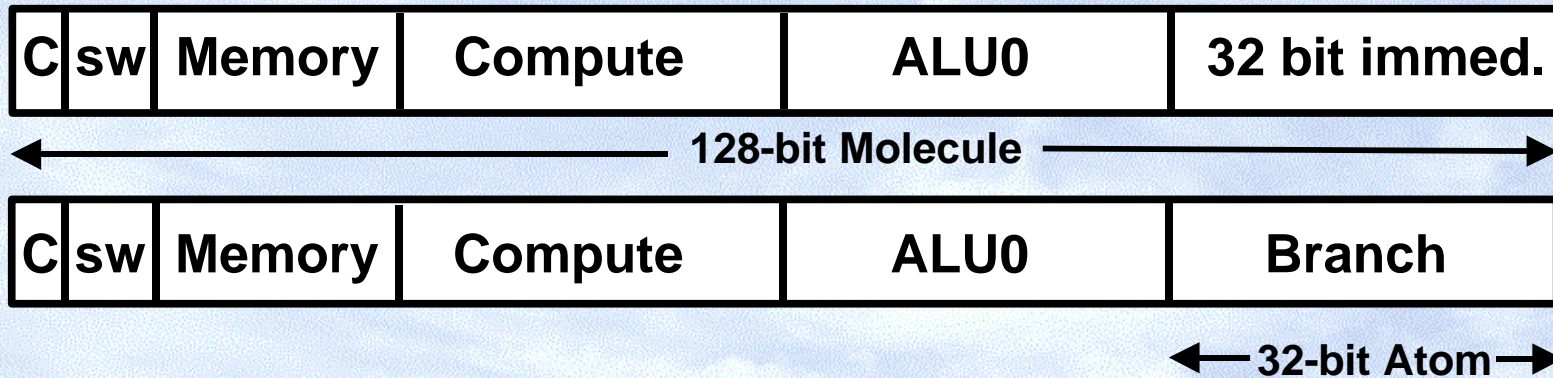
- ◆ Simple 6 Stage Pipeline
- ◆ Fetch0
- ◆ Fetch1
- ◆ Decode
- ◆ Register Read
- ◆ Execute
- ◆ Register Writeback

Code Morphing HW support

Integrated North Bridge

- ◆ PC133 SDR DRAM Controller
- ◆ DDR DRAM Controller
- ◆ PCI Bus Interface

Crusoe VLIW Processor Instruction Formats



C = 1 bit Commit instruction

sw = 2 bits for software use

Memory = Load or Store

**ALU operations are 3 address
register to register ops
with 64 general registers**

**Compute = ALU1, Floating Point
or Multimedia op**



Crusoe's Hardware Support for Dynamic Instruction Translation

or

How to make life easier for software

Traditional Code Generation Headaches

Life would be so much easier if it weren't for:

- Not enough registers
- Branches (basic blocks too short)
- Pointers
- Interrupts
- Faults
- Need to preserve original program order
- Self modifying code

Crusoe's hardware has support for all of the above

Basic Support

Lots of Registers

- 64 Integer Registers
- 32 Floating Point Registers

Lots of Cache

- 64 KB Level 1 Instruction Cache
- 64 KB Level 1 Data Cache
- 512 KB Level 2 combined I+D Cache
- 8 KB software managed local data memory
- 8 KB software managed local instruction memory

Simple pipeline

- 1 128-bit VLIW instruction (molecule) per clock
- Up to 4 atoms (RISC like op) per molecule
- So Software can tell the cost of an instruction!

Crusoe: Forgiveness and Time Travel

On typical procesors, code generation must be very conservative

- Must generate code that is “correct” in all cases
- Very rare circumstances ruin opportunities for optimization
 - e.g. memory aliases of two pointers
 - e.g. unexpected faults
- Less data in registers, more loads and stores
- Less speculation
- Result is less optimal code

Crusoe encourages aggressive optimization and speculation

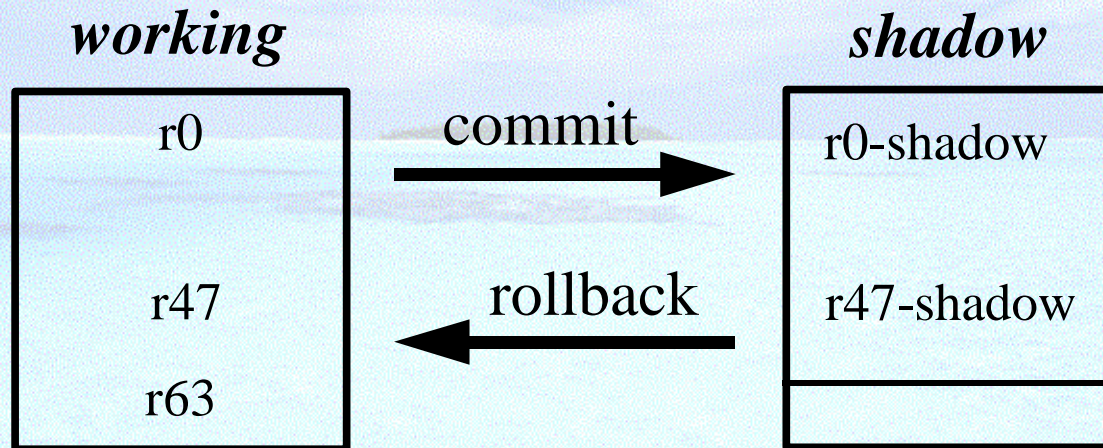
- If rare circumstances happen:
 - Give forgiveness
 - Travel back in time to before the exceptional event
 - Redo the event with more conservative code generation

Advanced Hardware Support

- ◆ Shadowed Register Files
- ◆ Gated Store Buffer
- ◆ Commit / rollback instructions
- ◆ Alias hardware
- ◆ Compare and Trap instructions
- ◆ Select instructions

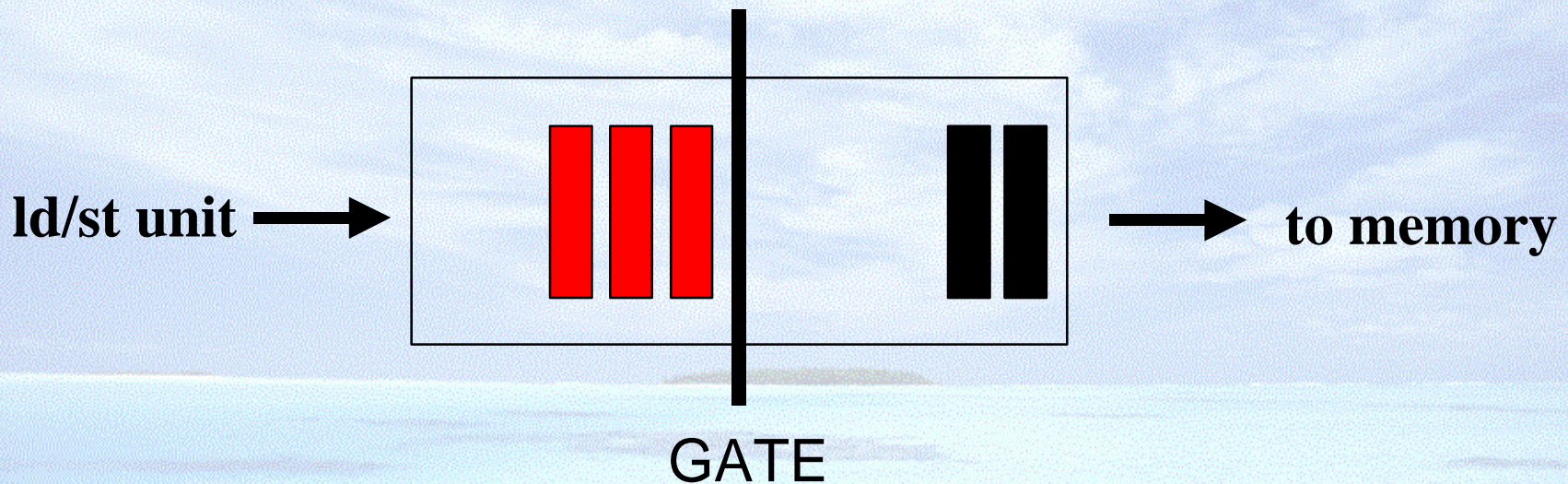
Shadowed Register Files

- ◆ Two copies of registers: *working* and *shadow*.
 - ◆ 64 Integer registers with 48 shadows
 - ◆ 32 Floating point registers with 16 shadows
- ◆ Normal atoms read/write working registers
- ◆ *commit* and *rollback* atoms copy working \leftrightarrow shadow



Gated Store Buffer

- ◆ Similar to traditional write buffer, except...
- ◆ *GATE* delays stores until next commit instr
- ◆ or Rollback instruction can undo stores



Load / Store Reordering

- ◆ Reordering register-register ALU ops is easy
- ◆ Reordering loads and stores: harder
- ◆ Need knowledge about memory addresses

st ..., [%x] ld %data, [%y]
ld %data, [%y] st ..., [%x]

The above transformation FAILS if pointers x equals y,
so this optimization is rarely used

Crusoe Alias Hardware

- ◆ Crusoe has unique Alias detection hardware
- ◆ Software uses special ld/st atoms when reordering
- ◆ Trap if memory regions overlap
- ◆ Ex: load protects memory, store checks
- ◆ Software can take corrective action, but rare

Original code (left):
`st ..., [%x]`
`ld %data, [%y]`

Transformed code (right):
`ldp %data, [%y]`
`stam ..., [%x]`

Correctness can be guaranteed, and easy to use

Crusoe Alias Hardware (e.g. #2)

Alias hardware can allow software to safely *eliminate* memory operations

`ld %r30,[%x]` \longrightarrow `ldp %r30,[%x]`

...

...

`st ...,[%y]` \longrightarrow `stam ...,[%y]`

`ld %r31,[%x]`

`use %r30`

`use %r31`

Correctness can be guaranteed, and easy to use

Multi-block Example

x86 (IA-32) Instruction Mix

```
1. movl %ecx,$0x3
2. jmp lbl1
lbl1:
3. movl %edx,0x2fc(%ebp)
4. movl %eax,0x304(%ebp)
5. movl %esi,$0x0
6. cmpl %edx,%eax
7. movl 0x40(%esp,1),$0x0
8. jle skip1
9. movl %esi,$0x1
skip1:
10. movl 0x6c(%esp,1),%esi
11. cmpl %edx,%eax
12. movl %eax,$0x1
13. jl skip2
14. xorl %eax,%eax
skip2:
15. movl %esi,0x308(%ebp)
16. movl %edi,0x300(%ebp)
17. movl 0x7c(%esp,1),%eax
18. cmpl %esi,%edi
19. movl %eax,$0x0
20. jnl exit1
exit2:
```

“Morphed” (128-bit) VLIW Instructions

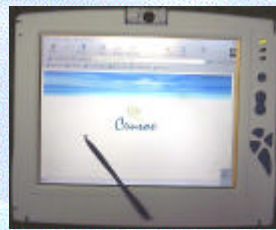
```
1. addi %r39,%ebp,0x2fc;commit
2. addi %r38,%ebp,0x304
3. ld %edx,[%r39]; add %r27,%r38,4; add %r26,%r38,-4
4. ld %r31,[%r38]; add %r35,0,1; add %r36,%esp,0x40
5. ldp %esi,[%r27]; add %r33,%esp,0x6c; sub.c %null,%edx,%r31
6. ldp %edi,[%r26]; sel #1e %r32,0,%r35;
7. stam 0,[%r36]; sel #1 %r24,%r35,0; add %r25,%esp,0x7c
8. stam %r32,[%r33];add %ecx,0,3; sub.c %null,%esi,%edi
9. st %r24,[%r25]; or %eax,0,0; brcc #lt,<exit2>
10. br <exit1>
```


Crusoe can be used in a Range of Mobile Computers

Crusoe in Mobile Internet Computers



Mobile Client



Web tablet

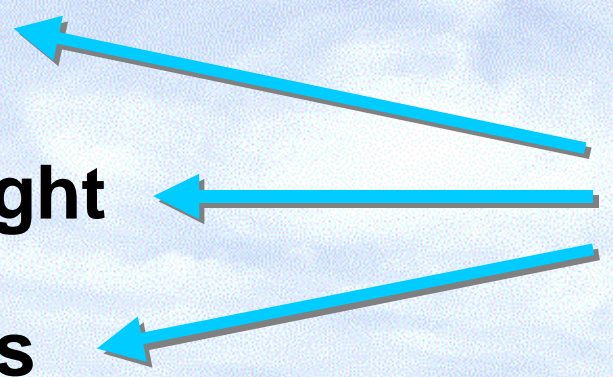


Thin+light Mobile PC



Full featured Mobile PC

Crusoe Offers Advantages for Mobile Internet Computing

- ◆ **Battery life**
 - ◆ **Size and weight**
 - ◆ **No noisy fans**
 - ◆ **Full PC software compatibility**
 - ◆ **LongRun power management**
- Low power is the key**
- 



Crusoe